SesameStreet++

There are 10 sorts of people in the world: those who understand binary *and those who don't.*
~ From "Proof Wiki" jokes page ~

Most of us think about "whole numbers" not too differently from the way we learned to count by watching Sesame Street, the difference being that now we can count a little higher. How we've trained ourselves it's automatic to think that the way we write a number or say a number IS the number.

If I say "I owe you 13 cents" and I give you one dime and three pennies then after thanking me profusely for repaying this staggering debt, we'll agree that it's settled with those coins equaling 13 pennies. We identify the symbol "13" very strongly with this particular number - it would be tough to get through life in the modern world without such an automatic process running in our brains. This example highlights what this particular symbol "13" actually means - one dime (1×10) plus three pennies (3×1).

Let's look at the number 13 in some alternative ways - it's the number of months in a year *plus* one month; what I'm suggesting is that there is no need for the symbol "13" in order to think about this particular number of months. Similarly, 13 is this many apples
🍏🍏🍏🍏🍏🍏🍏🍏🍏🍏🍏🍏🍏 ; or 13 is the sixth prime number (i.e., <1st: 2>, <2nd: 3>, <3rd: 5>, <4th: 7>, <5th: 11>, <6th: 13>); or perhaps 13 is "the number that follows the 2nd prime-number multiplied by the 1st prime-number squared", also written as $(3 \times 2^2) + 1$. None of these ways of thinking about the number 13 require that we represent it using the digits 1 and 3 butted up next to each other.

Each number exists independently from any symbol or word that might represent it. Numbers are an idea - perhaps such a strong idea that the universe wouldn't exist without it! Anyway, for our purposes whole numbers exist in some abstract realm - Each number is one whole unit more than the previous number, starting at nothing, that is "zero", and jumping to something, that is "one", then "one" more, which gets us to "two", then again to "three", etc. Continuing in this way forever… we get them all.

To get back to the idea of what a whole number *really* is, try to forget about the symbols or words we use and picture a pile of apples. There's zero apples (it's hard to show no-apples), then we introduce an "🍏" to get our very first, and smallest, non-empty pile of apples. Then add another apple to get a pile of "🍏🍏", then "🍏🍏🍏", then "🍏🍏🍏🍏" then some big pile of "🍏🍏🍏🍏…🍏🍏🍏" after we've been adding apples for a while. Each successively bigger pile of apples corresponds with each successive whole number.

We expand this entire set of whole numbers to include their negative-counterparts and call this larger set "integers". We denote the set of integers with this symbol: $\mathbb{Z}$. If we only want to talk about positive integers along with zero, we use this symbol: $\mathbb{Z}_{\geq 0}$. When we talk about numbers in

mathematics we always treat them as belonging to some kind of set (like the integers) in this pure, abstract sort of way.

However... using a "1" followed by a "3" to represent "🍏🍏🍏🍏🍏🍏🍏🍏🍏🍏🍏🍏🍏" is VERY handy. So we use Hindu-Arabic numbers and the positional notation of "base-ten", more commonly known as "decimal", to represent each specific integer. We slap a "-" on the front if we need to talk about a negative integer.

Base-ten representation of an integer is far superior to ancient Roman numerals for example. Try adding two numbers together in ancient Rome, or worse, multiplying or dividing them. What's XI times IX? Would you believe me if I told you it's XCIX? Unless you convert those to Hindu-Arabic numerals to check, you're just gonna have to trust me. Truth is - I don't know how to multiply using Roman numerals - nor did most Romans! Not only that, but I'll bet that most kids who graduated from Sesame Street can count higher than any Roman could - as the Roman system only effectively allowed counting up to 4999.

Using base-ten for us is automatic, we barely think about it when we're adding numbers or multiplying them - but it's worth looking carefully at how base-ten works - so let's examine it from the ground up.

It's useful to have simple symbols to represent each of the integers from one to nine, namely our familiar 1, 2, 3, 4, 5, 6, 7, 8 and 9 which have an interesting history and predate their use in base-ten.

Slightly more modern, but still quite ancient, is the symbol "0" for "zero", meaning "nothing". Zero also predates its use in base-ten but without zero, base-ten wouldn't be possible.

Base-ten uses the idea of stringing a series of digits together (a digit being one of the numbers 0, 1, 2, … 9), one after the other to be able to represent *any* whole number. Let's look at the first two-digit number, that is, ten, which as you well know looks like this: "10". This extra digit on the left tells us how many tens we have and the last, or rightmost digit says how many additional units to add to it.

So our very first two-digit number 10 means "one lot of ten - plus zero units". When we see "11" - we interpret it to mean "one lot of ten - plus one unit", and "12" is "one lot of ten - plus two units", etc. Continuing on; "20" - we interpret to mean "two lots of ten, plus zero units", etc. up to "90" meaning "nine lots of ten, plus zero units".

Following this line of reasoning since "10" now means the integer ten (i.e., 🍏🍏🍏🍏🍏🍏🍏🍏🍏🍏), then "100" must mean "ten lots of ten, plus zero units"- which is exactly what it means. We have a special word for this number we call it "one hundred" or "one lot of a hundred, plus zero lots of tens, plus zero units". Similarly "200" means "two lots of a hundred, plus zero lots of ten, plus zero units", etc., etc.

We can keep going by one-hundred until we similarly get to "1000" or "ten lots of a hundred, plus zero lots of ten, plus zero units" otherwise knows as "a thousand" or more specifically "one lot of a thousand, plus zero lots of a hundred, plus zero lots of ten, plus zero units".

It get's a little tedious to be so specific when reading out a number so our language has developed quite a few verbal shortcuts. Furthermore it doesn't take long before we run out of fancy names for these "powers of ten" like, million, billion, trillion, zillion etc.. So let's introduce some nice clean mathematical notation to describe these powers of ten and let's forget the fancy words.

$$100 = 10 \times 10 = 10^2 \ ,$$
$$1000 = 10 \times 10 \times 10 = 10^3 \ ,$$
$$10000 = 10 \times 10 \times 10 \times 10 = 10^4 \ ,$$
$$\cdots$$
$$10 \cdots 000 = 10 \times 10 \times 10 \times \cdots \times 10 = 10^k$$

The last line, $10^k = 10 \times 10 \times \cdots \times 10$, means there are k 10's multiplied together - also written as a 1 followed by k zeros. The above list explicitly shows the cases for $k = 2$, 3 and 4. Using the k like that is just a way to show that we can pick ANY whole number, i.e., there is no limit on how big k can be.

The notation of 10^k is very handy, in fact it extends to the case when $k = 0$ and $k = 1$. So 10^1 means that there is only one 10 multiplied together, in other words just the number 10. That $k = 1$ also tells us how many zero's follow the "1", so in this case there is one "0" following the "1" - again, our familiar 10.

How about when $k = 0$. Examining the pattern of how the power k relates to how many zeros follow the "1" (eg, 10, 100, 1000, etc.) it makes sense that $10^0 = 1$, i.e., no zeros follow the "1", which is exactly correct. Actually any number raised to the 0th power is 1, but we'll leave that discussion for another time.

Let's look at an example. Reading the number 46307 out according to our technique we can see that it's "four lots of ten-thousand, plus six lots of a thousand, plus three lots of a hundred, plus zero lots of ten, plus seven units":

$$
\begin{array}{rclcr}
& 4 \times 10000 & & & 40000 \\
+ & 6 \times 1000 & & + & 6000 \\
+ & 3 \times 100 & = & + & 300 \\
+ & 0 \times 10 & & + & 00 \\
+ & 7 \times 1 & & + & 7 \\
\hline
& & & = & 46307
\end{array}
$$

Written in terms of powers of ten: $46307 = 4 \times 10^4 + 6 \times 10^3 + 3 \times 10^2 + 0 \times 10^1 + 7 \times 10^0$.

You can think of each digit as being a little dial that controls how many lots of its corresponding power of ten will contribute to the value of the integer. **Claim**: Given that we can use as high as power of ten as we like and we can string together as LONG A LIST of digits as pleases us, that means that we can create ANY INTEGER WE WANT no matter how big it is.

That's a pretty tall claim. How do we know that we can create ALL the nonnegative integers with this scheme? For example, how do we know that we didn't miss one? Or how do we know that some string of digits doesn't represent two different integers? I know it seems silly to ask that, but attention to these kinds of details is what is referred to as "rigor" in Mathematics - it's necessary so we don't end up fooling ourselves or spouting bullshit - or if we are actually "full of it" then it's easy for other Mathematicians to call us on our nonsense.

We're going to jump into the deep end and make our claim in a careful mathy kind of way. Such a careful statement is called a theorem - stay tuned - theorems require proof, which we're going to supply!

Base-Ten Representation Theorem

Let $n, k \in \mathbb{Z}_{\geq 0}$: Then every n can be *uniquely* expressed as follows:

$$n = d_k 10^k + d_{k-1} 10^{k-1} + \cdots + d_2 10^2 + d_1 10^1 + d_0 10^0$$

for some k such that $0 \leq d_i \leq 9$ where $d_i, i \in \mathbb{Z}_{\geq 0}, 0 \leq i \leq k$.
Furthermore $d_k \neq 0$ except when $n = 0$.

Definition: n is represented in base-ten as $d_k d_{k-1} \cdots d_2 d_1 d_0$

A difficulty many folks have with math is the notation - it's kind of a language unto itself - like a computer program is a language. Let's take our theorem statement by statement and turn it into English.

i) "Let $n, k \in \mathbb{Z}_{\geq 0}$ "
This means we are going to talk about two distinct numbers that we are labelling n
and k. That strange looking $\in$ means "is an element of" (or "is a member of") and is always followed by something that is a "set". We talked above about the symbol $\mathbb{Z}_{\geq 0}$
which we defined as being the set of nonnegative integers. So, in other words, n can be one of 0 or 1 or 2 or 3 or ... any number - no matter how large - and the same goes for k.

ii) "Then every n can be uniquely expressed as follows"
···what we are about to say applies to ALL nonnegative integers and furthermore the statement is going to be *unique* for each integer.

iii) " $n = d_k 10^k + d_{k-1} 10^{k-1} + \cdots + d_2 10^2 + d_1 10^1 + d_0 10^0$ "
This is the expression in question. It equates n with a series of multiplications of some numbers (the d_i terms where i can stand in for any of numbers from 0 to k) times

descending powers of 10, and adds them all together.

It's useful to point out the meaning of our $d_0, d_1, d_2, \cdots, d_i, d_k$ etc. notation. Mathematical formulas such as this make judicious use of "subscripts" when coming up with names for lists of variables or constants. Subscripts following a letter or symbol, such as $d_0, d_1, d_2, \cdots$ are a handy way to get a list of variable or constant names that are similar looking to each other, and is meant to imply that they each fulfil a similar role to each other. Note here how the value of the subscript on each "d" corresponds to its power of ten, even in the single digit case when the subscript is 0, or the highest power case when the subscript is k.

iv) "for some k such that $0 \le d_i \le 9$ for each $d_i, i \in \mathbb{Z}_{\ge 0}, 0 \le i \le k$"
The "for some k" means that each integer n has one specific k associated with it.

It then states that those d_i terms are nonnegative integers, and can ONLY take on the values 0, 1, 2, 3, 4, 5, 6, 7, 8 or 9. Note that our uniqueness claim above means that each integer n has it's own unique list of d's.

It also is very fastidiously pointing out that the little 'i' we just introduced in the subscript of the d's is also an integer and can be as small as zero but only as large as our highest power k - whatever k might be. This is very picky stuff - like a computer program spelling things out very precisely so the computer knows exactly what you mean. (That's right - you are the computer).

v) "Furthermore $d_k \ne 0$ ..."
This is spelling out one more important fastidious detail. We want to make sure that the "most significant d", that is, our d_k that goes along with the highest power 10^k is **not** 0, in other words it must be one of 1, 2, 3, 4, 5, 6, 7, 8 or 9. This is necessary so that we can get our uniqueness property, otherwise we could say 13 = 013 = 0000013 which are all the integer 13, so let's outlaw this uninteresting and annoying possibility.

vi) "...except when $n = 0$"
completing that last statement which allows for *one exception* to the case where the "most significant digit" could be zero, and that's exactly when the integer n in question IS zero.

vii) "Definition: n is represented in base-ten as $d_k d_{k-1} \cdots d_2 d_1 d_0$"
This is introducing what it means to write the number out in base-ten; that is, we toss out all the extraneous stuff from our expression in (iii) above, and string all the "digits" one after another, from most significant digit d_k on the left down to least significant digit d_0 on the right.

Before we prove our theorem, consider that base-ten is not the only base in use these days. Since the introduction of the EDVAC computer, around 1950, there have been *many* orders of magnitude more calculations done in base-two (otherwise known as binary) by computers than have EVER been done by people in base-ten for the entirety of human history. (This might even be true if we only count one-day's worth of binary computer calculations - someone needs to check this!)

These binary-computer logic gates (the building blocks of the modern computer) can only take one of two states, that is; "off" or "on". We interpret these two states to represent these two numbers: 0 and 1. By doing so, in the same way that base-ten uses ten numbers 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9 for its digits; we can represent integers in base-two with just the digits 0 and 1. How is this possible? … first let's take an imaginary trip into deep space.

Consider distant Planet-Nova on which the emergent intelligent species only have nine fingers. They have three hands with three fingers each - anyway, they use base-nine, so they only use the numbers 0, 1, 2, 3, 4, 5, 6, 7 and 8 for their digits. So like we Earthlings do for the integer ten, instead of making up a new symbol for nine, they use "10" to represent the integer nine - which for them means "One lot of nine, plus zero units".

Similarly on Planet-Ocho, since they only have eight fingers, then they use base-eight and only use numbers 0, 1, 2, 3, 4, 5, 6 and 7 for their digits. For them "10" means "One lot of eight, plus zero units".

Finally we come upon Planet-Claire, where the poor bastards only have two fingers so they only use the digits 0 and 1 and base-two, so for them "10" means "one lot of two and zero units". So on Planet-Claire "10" means two. Recall above how we arrived at our 100 in base-ten, being "ten lots of ten, plus zero units" - similarly on Planet-Claire "100" in base-two for them means "Two lots of two plus zero units" in other words, four! What is "11" in base-two? Using our technique to describe the digits we see that it's "One lot of two, plus one unit", in other words three.

Here's how they count on Planet-Claire using base-two:

	base-two	base-ten		base-two	base-ten
	0	0		(…cont)	
	1	1		1001	9
+	1			1010	10
=	10	2		1011	11
+	1			1100	12
=	11	3		1101	13
+	1			1110	14
=	100	4		1111	15
+	1			10000	16
=	101	5		10001	17
+	1			…	
=	110	6		11111	31
+	1			100000	32
=	111	7		…	

+	1	1000000	64
= 1000	8	10000000	128
(cont...)		100000000	256 (...etc.)

Note something interesting in the list above - the powers of two, written in base-two, resemble our powers of 10 in base-ten! That is: $2^0(1) = 1$, $2^1(2) = 10$, $2^2(4) = 100$, $2^3(8) = 100$, $2^4(16) = 10000$, $2^5(32) = 100000$, $2^6(64) = 1000000$, $2^7(128) = 10000000$, $2^8(256) = 100000000$, etc.

Let's look at the binary number 11010 for example. Using our wordy technique to describe the number we can see that it's "One lot of sixteen, plus one lot of eight, plus zero lots of four, plus one lot of two, plus zero units":

$$
\begin{array}{rclcrcl}
 & 1 \times & 10000 & & & 10000 & (16) \\
+ & 1 \times & 1000 & & + & 1000 & (8) \\
+ & 0 \times & 100 & = & + & 000 & \\
+ & 1 \times & 10 & & + & 10 & (2) \\
+ & 0 \times & 1 & & + & 0 & \\
\hline
 & & & & = & 11010 & (26)
\end{array}
$$

Written in terms of powers of two: $11010 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$,

Each digit in base-two can be thought of as a little switch that turns on or off the contribution of its corresponding power of two. **Claim**: Given that the inhabitants of Planet-Claire can use as high a power of two as they like, and that they can string together as LONG A LIST of binary-digits as pleases them, that means that they can create ANY INTEGER THEY WANT no matter how big it is.

Sound familiar? Let's restate our theorem for base-ten but rewritten for base-two.

Base-Two Representation Theorem

Let $n, k \in \mathbb{Z}_{\geq 0}$: Then every n can be uniquely expressed as follows:

$$n = d_k 2^k + d_{k-1} 2^{k-1} + \cdots + d_2 2^2 + d_1 2^1 + d_0 2^0$$

for some k such that $0 \leq d_i \leq 1$ where $d_i, i \in \mathbb{Z}_{\geq 0}, 0 \leq i \leq k$.
Furthermore $d_k \neq 0$ except when $n = 0$.

Definition: n is represented in base-two as $(d_k d_{k-1} \cdots d_2 d_1 d_0)_2$

Before we go on, I want to introduce a little notation to help avoid confusion. How do you know what I'm talking about if I just write "1000"? Do I mean 10^3 or 2^3 ? If there is any possibility for confusion we write the number like this $(1000)_{10}$ for the base-ten version of 1000 and $(1000)_2$ to mean the binary version.

As is hinted by the habits of our various alien friends above it seems that we can use ANY integer greater than or equal to 2 as a base. In fact computer graphics artists are known to stumble upon numbers written in hexadecimal, which is base 16 (usually relating to specifying a color-channel). Base 16 introduces some new single-character symbols to the usual numbers 1, 2, 3, 4, 5, 6, 7, 8 and 9, to represent each number 10, 11, 12, 13, 14 and 15, namely: A, B, C, D, E and F where $A_{16}=(10)_{10}$, $B_{16}=(11)_{10}$, $C_{16}=(12)_{10}$, $D_{16}=(13)_{10}$, $E_{16}=(14)_{10}$, $F_{16}=(15)_{10}$. So if you see this number $(80FB)_{16}$ then I bet at this point (if you take a little time with a calculator and a pad of paper and pencil) then you can figure out that it's $(33019)_{10}$. Note that if we omit the parentheses and subscript from a number, it means we're talking about it in base-ten - our "default" base. Case in point… the subscripts that we use to denote the base (like the "16" in $(80FB)_{16}$) are written in base-ten!

We really need to get on with proving our two theorems above. But what about proving the "base-nine" version of the theorem for the aliens on Planet-Nova, or the "base-eight" version for the inhabitants of Planet-Ocho?

To cover all bases (pun intended) let's restate our theorem for the general case, call it "base b", where b is any number greater than one. If we can prove *that* theorem, then we'll automatically get *all* the cases of specific bases for free.

Basis Representation Theorem

Let n , k, $b \in \mathbb{Z}_{\geq 0}$ such that $b \geq 2$: Then every integer n can be uniquely expressed as follows:

$$n = d_k b^k + d_{k-1} b^{k-1} + \cdots + d_2 b^2 + d_1 b^1 + d_0 b^0$$

for some k such that $0 \leq d_i \leq (b-1)$ where $d_i, i \in \mathbb{Z}_{\geq 0}$, $0 \leq i \leq k$.
Furthermore $d_k \neq 0$ except when $n = 0$.

Definition:
n is represented in base-b as $(d_k d_{k-1} \cdots d_2 d_1 d_0)_b$
and each d_i is a base-b-digit of n.

The motivation for our proof is to spell out the technique for actually constructing the base-b representation of any integer and show that this can ALWAYS be done, and each step is unique.

This technique makes repeated use of a well established theorem which is called the "Euclidean Division Theorem" - It sounds onerous, but don't worry, you all learned it in the third grade but perhaps not so formally. It simply states the following:

Euclidean Division Theorem

For all $a, b \in \mathbb{Z}_{\geq 0}$ where $b \neq 0$, there exists unique integers q, r such that:

$$a = qb + r \quad \text{and} \quad 0 \leq r \leq (b-1)$$

Definition: In the above equation:

> a is the **dividend**
> b is the **divisor**
> q is the **quotient**
> r is the **remainder**

This is how you first learned to divide. For example if someone asks you "What is nineteen divided by three?" - you'd answer "six with one remaining". Here 19 is the dividend, 3 is the divisor, 6 is the quotient and 1 is the remainder. Written in the form of our theorem:

$$19 = 6 \times 3 + 1$$

An important thing to note here is that the quotient and the remainder are **unique** for each pair of dividend and divisors as long as the divisor b is not equal to zero. We won't prove the Euclidean Division Theorem - we'll take it as already proven and use it.

Proof of "Basis Representation Theorem"

Let $n, b \in \mathbb{Z}_{\geq 0}$ such that $b \geq 2$.

We are going to construct a "base b" representation of n.

Let n be a dividend and since $b \neq 0$ we can use b as a divisor, then by the Euclidean Division Theorem there exists a unique quotient $q_1 \geq 0$ and remainder d_0 such that:

$$n = q_1 b + d_0 \quad \text{where } 0 \leq d_0 \leq (b-1)$$

Suppose $n < b$

> $n < b$ is true if and only if $q_1 = 0$ and $d_0 = n$ in which case:
>
> $$
> \begin{aligned}
> n \ &= 0b + n && \text{(Eucl. Div. Thm.)} \\
> &= n \cdot 1 && \text{(Axiomatic: multiply by "identity" 1)} \\
> &= n b^0 && \text{(Rewrite 1 as power of } b\text{)} \\
> &= d_0 b^0 && \text{(Substitute } d_0 \text{ for } n \text{ and note: } 0 \leq d_0 \leq (b-1)\text{)}
> \end{aligned}
> $$
>
> Therefore since n itself satisfies the criteria for a single digit in a base-b representation of n, then we've proven the "Basis Representation Theorem" when n is less than b.

Suppose $n \geq b$

Now consider q_1 as the dividend keeping b as the divisor, then by the Euclidean Division Theorem there exist unique quotient $q_2 \geq 0$ and remainder d_1 such that:

$$q_1 = q_2 b + d_1 \quad \text{where } 0 \leq d_1 \leq (b-1)$$

Similarly by applying the Euclidean Division Theorem to the new quotient q_2 and continuing ad infinitum we can say that there exist unique nonnegative integer quotients $q_3, q_4, \cdots, q_i, q_{i+1}, \cdots$ and remainders $d_2, d_3, \cdots, d_i, \cdots$ such that:

$$q_2 = q_3 b + d_2 \quad \text{where } 0 \leq d_2 \leq (b-1)$$
$$q_3 = q_4 b + d_3 \quad \text{where } 0 \leq d_3 \leq (b-1)$$
...
$$q_i = q_{i+1} b + d_i \quad \text{where } 0 \leq d_i \leq (b-1)$$
...

We know that q_1 is non-zero (otherwise we'd be in the case that $n < b$) but suppose that one of our quotients in the list is zero, say $q_i = 0$, then *all* subsequent quotients, $q_{i+1}, q_{i+2}, \cdots$ must also be zero because:

$$q_i = 0 = q_{i+1} b + d_i = 0 \cdot b + 0$$

since $b \neq 0$ then both q_{i+1} and d_i **must be** zero since by definition neither of them can take on negative values (which would allow for something like $(qb - d) = 0$).

This same argument can be repeated to show that $q_{i+2}, q_{i+3}, \cdots$, must also be zero.

Now we'll show that there MUST be a quotient in the list that is zero.

We start by noting that for any quotient q_i that $q_i \geq q_{i+1}$ because:

$$
\begin{aligned}
q_i &= q_{i+1} b + d_i && \text{(by definition)} \\
&\geq q_{i+1} b + 0 && \\
&\geq q_{i+1} \times 2 && \text{(since } b \geq 2) \\
&\geq q_{i+1} \times 1 && \text{(*)} \\
&= q_{i+1}
\end{aligned}
$$

Suppose $q_{i+1} \neq 0$ then the inequality $q_i \geq q_{i+1}$ can be rewritten as a strict inequality thusly:

$$q_i > q_{i+1}$$

because at (*), $2q_{i+1} \geq q_{i+1}$ has equality if and only if $q_{i+1} = 0$

Furthermore when $q_i > 0$ and $q_{i+1} = 0$ then $q_i > q_{i+1}$ is trivially true.

We know that $q_1 > 0$ so as long as no subsequent q_i is zero we can write

$$q_1 > q_2 > q_3 > \cdots > q_i > \cdots$$

Which implies that one of the quotients must eventually be zero, for even if each quotient was only one unit smaller than the previous quotient then we are guaranteed that the q_1^{th} quotient must be zero. In other words:

$$q_1 > (q_1 - 1) > (q_1 - 2) > (q_1 - 3) > \cdots > (q_1 - q_1) = 0$$

[Aside: The series of quotients will ultimately reach the value zero *much* faster than the previous statement implies - indeed it will only take at most $\lfloor log_b(q_1) \rfloor + 1$ steps which is *staggeringly* faster than q_1 steps. Eg. $(\lfloor log_{10}(789456123) \rfloor + 1) = 9$ which is a considerably smaller number than 789,456,123 !]

Let k be the number of nonzero quotients in our list above. Then $q_{k+1} = 0$ and our complete list of quotients is as follows:

$$q_1 = q_2 b + d_1$$
$$q_2 = q_3 b + d_2$$
$$\dots$$
$$q_{k-1} = q_k b + d_{k-1}$$
$$q_k = d_k$$

---wip jpr ---

Part of our proof is going to make use of a powerful 2-step technique - namely "induction". We start by making some assertion about a number *n*. This assertion could be anything including the theorem itself. Then,

1. We show that the assertion is true for the "base case", when $n = 1$;
2. We then we *assume* that the proposed assertion is true for some arbitrary number *n*, and using *this* fact (as well as other established properties of numbers like the axioms, or other previously proven theorems etc.) we show that the proposed assertion must also be true for the *next* larger whole number $n + 1$.

If we can do those two things, then we've proven the assertion FOR ALL whole numbers.

Why would this approach actually prove an assertion for *all* whole numbers? Suppose P(*n*) - means "some property of the whole number *n*". If we accept the following statement as true: "If P(*n*) is true, then P(*n*+1) is true" then if we also know that "P(1) is true" - those two statements together imply that P(2) is true, which implies that P(3) is true, which in turn leads to the truth of P(4), etc., over and over so that we eventually reach ANY number.

Some people picture an infinite row of dominoes. Step 1 is like being able to knock over the first domino. Step 2 is like the fact that any one domino has the ability to knock over the next. Once you've knocked over the first domino, they all fall.

Often proofs make use of little mini-theorems of their own. Creating these mini-theorems is way to simplify a step in the main proof by establishing a useful non-trivial intermediary result. It makes reading the main proof easier to follow by not having us get side tracked by technicalities. These mini-proofs are called "Lemmas" and we're going to make one right now because we're going to need it later.

Lemma

Let $k \in \mathbb{Z}$ such that $k \geq 1$
Also let $b, d_0, \cdots, d_k, n_0, \cdots, n_k \in \mathbb{Z}_{\geq 0}$, such that

$n_0 = bn_1 + d_0$, and
$n_1 = bn_2 + d_1$, and
...
$n_{k-1} = bn_k + d_{k-1}$, and
$n_k = bn_{k+1} + d_k$

then:
$$n_0 = n_{k+1}b^{k+1} + d_k b^k + d_{k-1}b^{k-1} + \cdots + d_2 b^2 + d_1 b^1 + d_0 b^0$$

Our lemma defines a list of expressions each one referring to the next expression. Each expression is labeled with the name n_i for the ith expression which would look like this: $n_i = bn_{i+1} + d_i$

To explain the motivation for this lemma, imagine starting at the last expression that is called n_k and substituting it into the previous expression for n_{k-1}, then take the new expression for n_{k-1} and substitute it into the expression for n_{k-2}, etc. After continuing that process of substitution all the way back to n hopefully you can see that you'll end up with this nested expression:

$$n = b(b(b\cdots b(bn_{k+1} + d_k) + d_{k-1}) + \cdots + d_2) + d_1) + d_0$$

It's a bit of a stretch of the imagination to ask you to just accept that expanding this nested expression gives us $n = n_{k+1}b^{k+1} + d_k b^k + d_{k-1}b^{k-1} + \cdots + d_2 b^2 + d_1 b^1 + d_0 b^0$, so since we are going to need to make just such a logical leap later on in our Proof of the "Unique Representation of Integers in an Arbitrary Base" theorem, then to make things nice and sparkling clear, we create our lemma and prove it.

Proof of lemma

This proof will make use of induction on k (which is the subscript on the names of each expression, i.e., "n_k").

Prove for $k = 0$

$$n = bn_1 + d_0 = n_1 b + d_0 \cdot 1 = n_1 b^1 + d_0 b^0$$

which is trivially true by the definition of n and a valid expression even when $b = 0$, which means we'd have a 0^0 in there. Mathematicians define 0^0 to be 1 to simplify annoying cases exactly like this one! (Google it - it's legit!)

Now we assume the lemma is true for $k = i$

Then if we have a list of expressions:

$$n = bn_1 + d_0,$$
$$\ldots$$
$$n_i = bn_{i+1} + d_i$$

for some $b, d_0, \cdots d_i, n_{i+1} \in \mathbb{Z}_{\geq 0}$

then the following is true:

$$n = n_{i+1} b^{i+1} + d_i b^i + d_{i-1} b^{i-1} + \cdots + d_2 b^2 + d_1 b^1 + d_0 b^0$$

and we will show that the lemma is true for $k = i + 1$

In other words, if we extend our list of expressions by one more item, that our new expanded expression for n will similarly extend out one more term.

Let $d_{i+1}, n_{i+2} \in \mathbb{Z}_{\geq 0}$, then n_{i+1} can always be written out as:

$$n_{i+1} = bn_{i+2} + d_{i+1}$$

For some choice of d_{i+1} and n_{i+2}, for example it's always possible if we let $d_{i+1} = n_{i+1}$ and $n_{i+2} = 0$, because

$$n_{i+1} = 0 + n_{i+1} = b \cdot 0 + n_{i+1} = bn_{i+2} + d_{i+1}$$

but there are other possibilities for the choices of d_{i+1} and n_{i+2} when $b < n_{i+1}$.

The important point is that it is always possible express the number n_{i+1} with an expression using newly defined terms such that it takes the form of all the other expressions in our list. Therefore,

$$
\begin{aligned}
n &= (n_{i+2} b + d_{i+1}) b^{i+1} + d_i b^i + d_{i-1} b^{i-1} + \cdots + d_2 b^2 + d_1 b^1 + d_0 b^0 \\
&= n_{i+2} b b^{i+1} + d_{i+1} b^{i+1} + d_i b^i + d_{i-1} b^{i-1} + \cdots + d_2 b^2 + d_1 b^1 + d_0 b^0 \\
&= n_{i+2} b^{i+2} + d_{i+1} b^{i+1} + d_i b^i + d_{i-1} b^{i-1} + \cdots + d_2 b^2 + d_1 b^1 + d_0 b^0
\end{aligned}
$$

which extends our resulting expression by one extra term proving the lemma.

QED

----old ending follows---

Ok we're ready to finally jump in. In order to prove our theorem - first we will prove that there is such a representation for all integers n (existence). Meaning that every integer has a way of being written in the form described by the theorem - especially as relates to the restrictions on the values that the "digits" can take on. After that has been established we will use another technique to prove that each such representation is unique - in other words there aren't two (or more) ways to represent the same integer in our base.

Existence Proof (by induction on n):

Base case:

For clarity, let's prove both $n = 0$ and $n = 1$. (This is like showing that we can knock over both the 1st and the 2nd dominoes.)

Let $n = 0$. We can choose $k = 0$ and $d_0 = 0$. (This is the one exception spelled out in the theorem in which the most significant digit of n is allowed to be zero.) Then,

$$n = d_0 b^0 = 0 \times b^0 = 0,$$

showing that we have a valid representation for 0 in base b since our only digit $d_0 = 0 \leq (b - 1), \ \forall \ b \geq 2$.   (The " $\forall$ " symbol is shorthand for "for all".)

Now Let $n = 1$. In this case, we can choose $k = 0$ and $d_0 = 1$. Then,

$$n = d_0 b^0 = 1 \times b^0 = 1 \times 1 = 1,$$

showing that we have a valid representation for 1 in base b since $d_0 = 1 \leq (b - 1), \ \forall \ b \geq 2$.

$n+1^{st}$ step:

Assume that n has a valid representation in base b. That is, we assume that n can be written thusly:

$$n = d_k b^k + d_{k-1} b^{k-1} + \cdots + d_2 b^2 + d_1 b^1 + d_0 b^0$$

with all the appropriate conditions holding for the values of d_i and b and k; and we will prove that $n + 1$ also has a valid representation in base b.

We're going to break this step into two cases which between the two covers all possibilities.

Case 1) $0 \le d_0 \le (b - 2)$

> This case looks at the situation where the *least significant digit* of n is strictly-less-than the LARGEST value it can take in base b.
>
> For example, in base-two, d_0 can only be 0. In base-five, d_0 can be at most 3. In base-ten, d_0 can be at most 8, etc.
>
> So we can express $n + 1$ as follows:
>
> $$
> \begin{aligned}
> n + 1 &= (d_k b^k + d_{k-1} b^{k-1} + \cdots + d_2 b^2 + d_1 b^1 + d_0 b^0) + 1 \\
> &= d_k b^k + d_{k-1} b^{k-1} + \cdots + d_2 b^2 + d_1 b^1 + d_0 b^0 + 1 \times 1 \\
> &= d_k b^k + d_{k-1} b^{k-1} + \cdots + d_2 b^2 + d_1 b^1 + d_0 b^0 + 1 \times b^0 \\
> &= d_k b^k + d_{k-1} b^{k-1} + \cdots + d_2 b^2 + d_1 b^1 + (d_0 + 1) b^0
> \end{aligned}
> $$
>
> and since $d_0 \le (b - 2)$ (our assumption for "case 1"), then
>
> $$(d_0 + 1) \le (b - 2) + 1 \le (b - 1),$$
>
> therefore the "least significant digit" (i.e. $(d_0 + 1)$ which corresponds the lowest power b^0) of $n + 1$ is less than or equal to $(b - 1)$, which means it is a valid digit in base b.
>
> Since all other d_i terms (i.e. $d_1, d_2, \cdots, d_k$) for $n + 1$ are unchanged from their representations for n and n is assumed to be a valid representation in base b then **all** the digits of $n + 1$ are valid in base b.
>
> Therefore we've established the truth of "Case 1" for the integer $n + 1$.

Case 2) $d_0 = (b - 1)$

> Now we'll look at the case when the least significant digit of n is equal to the largest value it can take in base b, that is, $d_0 = (b - 1)$. (Note that between "Case 2" here and "Case 1" above, we're covering *all* the possibilities for what d_0 can be.)
>
> For example in base-two, $d_0 = 1$; in base 5, $d_0 = 4$; in base-ten $d_0 = 9$, etc.
>
> Let's temporarily introduce a new digit, $d_{k+1} = 0$ to our representation of n. The motivation being that in the situation where we're adding 1 to 999, that we need another "digit" to "carry" the 1 over into so that we can get 1000.

$$\begin{aligned} n &= d_k b^k + d_{k-1} b^{k-1} + \cdots + d_2 b^2 + d_1 b^1 + d_0 b^0 \\ &= 0 + d_k b^k + d_{k-1} b^{k-1} + \cdots + d_2 b^2 + d_1 b^1 + d_0 b^0 \\ &= 0 \cdot b^{k+1} + d_k b^k + d_{k-1} b^{k-1} + \cdots + d_2 b^2 + d_1 b^1 + d_0 b^0 \\ &= d_{k+1} b^{k+1} + d_k b^k + d_{k-1} b^{k-1} + \cdots + d_2 b^2 + d_1 b^1 + d_0 b^0 \end{aligned}$$

Also, let $j \in \mathbb{Z}_{\geq 0}$ be the lowest power of b such that $d_j < (b-1)$, $(k+1) \geq j \geq 1$, meaning we write n as follows for some j.

$$n = d_{k+1} b^{k+1} + \cdots + d_j b^j + (b-1) b^{j-1} + \cdots + (b-1) b^1 + (b-1) b^0$$

For example, if $n = 69412999$, then $j = 3$, since 10^3 is the lowest power of 10 such that its digit $d_3 = 2$ is less than 9.

Let's get picky for a minute and think carefully about some boundary conditions, to make sure that our new expression for n is always possible to write this way.

What about when $k = 0$ meaning that n is only one digit long? Since we specified that $(k+1) \geq j \geq 1$ or substituting zero for k then $(0+1) = 1 \geq j \geq 1$, which means that j must be 1. Note that we tacked on the extra digit d_{k+1} above, and by substituting $k = 0$ we see that $d_{k+1} = d_{0+1} = d_1 = 0$. Recall that we are in "Case 2" of our proof, so we know that $d_0 = (b-1)$ so it's true that b^1 is the lowest power of b such that $d_1 = 0 < (b-1)$, $\forall\ b \geq 2$. So we're OK here!

What about the case when ALL the digits of n, namely $d_0,\ d_1, \cdots, d_k$ are equal to $(b-1)$? (For example $n = 999999$) Then $j = (k+1)$, and $d_j = d_{k+1} = 0 < (b-1)$.

Good! We're covered in terms of our new expression for n being valid for all possible values of j i.e.; $1 \leq j \leq (k+1)$. So let's do some algebraic manipulations on n.

$$\begin{aligned} n &= d_{k+1} b^{k+1} + \cdots + d_j b^j + (b-1) b^{j-1} + \cdots + (b-1) b^1 + (b-1) b^0 \\ &= d_{k+1} b^{k+1} + \cdots + d_j b^j + (b-1)(b^{j-1} + \cdots + b^1 + b^0) \\ &= d_{k+1} b^{k+1} + \cdots + d_j b^j + b(b^{j-1} + \cdots + b^1 + b^0) - (b^{j-1} + \cdots + b^1 + b^0) \\ &= d_{k+1} b^{k+1} + \cdots + d_j b^j + (bb^{j-1} + \cdots + bb^1 + bb^0) - (b^{j-1} + \cdots + b^1 + b^0) \\ &= d_{k+1} b^{k+1} + \cdots + d_j b^j + (b^j + \cdots + b^2 + b^1) - b^{j-1} - \cdots - b^1 - b^0 \\ &= d_{k+1} b^{k+1} + \cdots + d_j b^j + b^j + (b^{j-1} - b^{j-1}) + \cdots + (b^2 - b^2) + (b^1 - b^1) - b^0 \\ &= d_{k+1} b^{k+1} + \cdots + (d_j + 1) b^j - b^0 \\ &= d_{k+1} b^{k+1} + \cdots + (d_j + 1) b^j - 1 \end{aligned}$$

Therefore

$$n + 1 = d_{k+1}b^{k+1} + \cdots + (d_j + 1)b^j - 1 + 1$$
$$= d_{k+1}b^{k+1} + \cdots + (d_j + 1)b^j$$

Since we picked j such that $d_j < (b - 1)$, less restate the inequality as $d_j \leq (b - 2)$ therefore,

$$(d_j + 1) \leq (b - 2) + 1 \leq (b - 1),$$

meaning the digit $(d_j + 1)$ for the base b representation of $n + 1$ is valid in base b.

All digits d_{j+1} and higher remain unchanged from what they were for the base b representation of n, and all digits d_{j-1} and lower are 0. Therefore all the digits of the base b representation of $n + 1$ are valid in base b.

To cleanly button this up we note that if we were dealing with the case that $j = k + 1$, (eg. n was something like 99999) then we can see that $n + 1 = b^{k+1}$ because:

$$n + 1 = (d_j + 1)b^j$$
$$= (d_{k+1} + 1)b^{k+1}$$
$$= (0 + 1)b^{k+1}$$
$$= b^{k+1}$$

Meaning that $n + 1$ has a $(k+1)^{\text{st}}$ digit and it's equal to 1, with all the rest of the digits following being 0.

In all other cases (i.e.; $j < (k + 1)$) we discard the extra "temporary" digit $d_{k+1} = 0$ that we created from our expression, and are left with an expression for $n + 1$ that valid for base b.

QED - existence proof

Uniqueness Proof for *n*:

Let's assume that our base b representation for n is not unique, Showing that this assumption leads to a logical contradiction, then the only possibility remains is that our assumption was wrong, thereby proving uniqueness.

Suppose n is not unique and that,

$$n = d_k b^k + d_{k-1}b^{k-1} + \cdots + d_2 b^2 + d_1 b^1 + d_0 b^0,$$

also suppose,

$$n = c_k b^k + c_{k-1}b^{k-1} + \cdots + c_2 b^2 + c_1 b^1 + c_0 b^0$$

Let's further suppose that the index j is the first power such that the digits $d_j \neq c_j$ and without any loss of generality, let's assume that $d_j > c_j$.

Therefore